



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/418,943	10/15/1999	TIMOTHY CHARLES SOWELL	202231	9508
7590 05/29/2009				
MARK JOY LEYDIG VOIT & MAYER LTD TWO PRUDENTIAL PLAZA SUITE 4900 180 NORTH STETSON CHICAGO, IL 606016780				
EXAMINER				
NGUYEN, NGA B				
ART UNIT		PAPER NUMBER		
3692				
MAIL DATE		DELIVERY MODE		
05/29/2009		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte TIMOTHY CHARLES SOWELL

Appeal 2009-000451
Application 09/418,943
Technology Center 3600

Decided:¹ May 29, 2009

Before HUBERT C. LORIN, LINDA E. HORNER, and
ANTON W. FETTING, *Administrative Patent Judges*.

LORIN, *Administrative Patent Judge*.

DECISION ON APPEAL

An oral hearing was held on May 14, 2009.

¹ The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, begins to run from the decided date shown on this page of the decision. The time period does not run from the Mail Date (paper delivery) or Notification Date (electronic delivery).

STATEMENT OF THE CASE

Timothy Charles Sowell (Appellant) seeks our review under 35 U.S.C. § 134 of the non-final rejection of claims 1-12, 14-44, 53-55, and 74-77. Claims 45-52 and 56-73 have been withdrawn and claim 13 has been canceled. We have jurisdiction under 35 U.S.C. § 6(b) (2002).

SUMMARY OF DECISION

We AFFIRM-IN-PART.²

THE INVENTION

The invention is directed to charging customers for software distributed to them according to their use of the software.

Claim 1, reproduced below, is illustrative of the subject matter on appeal.

1. A method for charging customers for use of software comprising the steps of:
 - establishing a use-based pricing scheme for a set of software modules;
 - distributing the set of software modules to a customer, wherein the set of software modules comprise at least one object class from which objects are instantiated on a customer system;
 - monitoring customer use of the software modules; and
 - charging the customer according to use of the distributed software modules as determined during the monitoring step, wherein software usage is measured according to object instances created from the at least one object class.

² Our decision will make reference to the Appellants' Appeal Brief ("App. Br.," filed Dec. 7, 2004, reinstated Apr. 17, 2007) and Reply Brief ("Reply Br.," filed Oct. 10, 2007), and the Examiner's Answer ("Answer," mailed Aug. 10, 2007).

THE REJECTIONS

The Examiner relies upon the following as evidence of unpatentability:

Archibald	US 5,825,883	Oct. 20, 1998
Ahmad	US 5,925,127	Jul. 20, 1999
Knapton, III	US 6,363,486 B1	Mar. 26, 2002
Sobeski	US 6,499,035 B1	Dec. 24, 2002

The following rejections are before us for review:

1. Claims 1-3, 5-7, 9-12, 14-25, 27-35, 37-40, 42-44, 53-55, and 74-77 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Archibald, Knapton, III, and Sobeski.
2. Claims 4, 8, 26, 36, and 41 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Archibald, Knapton, III, Sobeski, and Ahmad.

ARGUMENTS

According to the Examiner, Archibald discloses all of the claimed subject matter but for the “the set of software modules compris[ing] at least one object class from which objects are instantiated on a customer system and software usage is measured according to object instances created from the at least one object class.” Answer 3. The Examiner relied on Knapton, III, for its disclosure of “software modules executed on the customer computer [that] is developed using object oriented programming” (Answer 3-4) and Sobeski for its disclosure of “software modules executed on the customer computer system . . . developed using Java objects [which] comprise at least one object class from which objects are instantiated on a

customer system” (Answer 4) to overcome the shortcomings of Archibald. The Examiner also stated that “Sobeski discloses ... software usage [that] is measured according to object instances created from the at least one object class (column 6, line 47 - column 7, line 60.)” Answer 4. The Examiner concluded that Archibald, Knapton, III, and Sobeski disclose all the limitations of the subject matter claimed and the combination of Archibald, Knapton, III, and Sobeski would lead one of ordinary skill in the art to the claimed subject matter. *Id.*

The Appellant argued that the references do not disclose charging customers based on the monitored use of created object instances.

The existence of object-oriented programming (Knapton) and regulating/licensing object usage on client systems (Sobeski) at the time Appellant conceived the current invention does not overcome the shortcomings of the Archibald reference and therefore does not support the obviousness conclusion reached within the Office Action. Appellant’s claimed invention goes beyond merely reciting that customers are charged for using applications that contain objects. Instead, the claims recite that object instances are created from object classes that are distributed to customer systems. The customers are thereafter *charged in accordance with the monitored use of the created object instances*.

App. Br. 9-10.

The Examiner responded by repeating, in part, that “Sobiski [*sic.*, Sobeski] discloses ... software usage [that] is measured according to object instances created from the at least one object class (column 6, line 47- column 7, line 60.)” Answer 15.

The Appellant replied, in part, that “Sobeski does not teach monitoring usage of software modules through created objects.” Reply Br. 5. “[T]he combined teachings of Sobeski, Knapton III, and Archibald do

not, in combination, disclose or suggest the recited invention that is directed to *charging a customer based upon monitored object instances created from software modules* on the consumer's computer system." Reply Br. 6.

ISSUES

There are two issues.

The first issue is: do the cited references disclose or suggest to one of ordinary skill in the art measuring software usage according to object instances created from the at least one object class as claimed? This issue pertains only to method claims 1-12, 14-44, and 74-77, as those are the claims which include the measuring step.

The second issue is: are claims 53-55 patentable over the cited prior art? These claims are directed to a "memory."

FINDINGS OF FACT

We find that the following enumerated findings of fact (FF) are supported by at least a preponderance of the evidence. *Ethicon, Inc. v. Quigg*, 849 F.2d 1422, 1427 (Fed. Cir. 1988) (explaining the general evidentiary standard for proceedings before the Office).

Claim construction

1. The independent claims are method claims 1, 31, 37, and 42, and claim 53 to a "memory."
2. All the method claims include a step of charging customers according to the use of distributed software modules wherein the software usage "is measured according to object instances created from the at least one object class."

3. Claim 53 describes a “memory” comprising “an executable program module including at least one object class from which instantiated objects are rendered.”
4. A “software module” is “a data construct including at least an identifiable, integration-capable set of data and/or program instructions. An example of a software module is a data construct including an object class or set of classes from which objects are created and executed.” Specification 5:24-27. A software module may be a self-contained program. *See* Specification 5:28.
5. An executable program module is a software module. *See the* Specification 5:28 (software modules “include executables such as self-contained programs”).
6. An “object class” is a “set of building blocks and templates” incorporated in software systems. Specification 3:7-8.
7. An “instance” is “a copy of data and/or program code derived/copied from a software module.” Specification 6:3-6. “An example of an instance is an object created from an object class in a software module.” Specification 6:4-5.
8. The Specification does not appear to provide a definition for the phrase “instantiated objects” (claim 53).
9. “Instantiation” has had the following definition:

In object-oriented programming, to produce a completed object by filling in values in place of variables in a class template.

Webster’s New World Dictionary of Computer Terms 279 (8th Ed. 2000).(Entry for “instantiation.”)
10. “Memory” has had the following definition:

The computer's primary storage, such as random access memory (RAM), as distinguished from its secondary storage, such as disk drives.

Webster's New World Dictionary of Computer Terms 340 (8th Ed. 2000.) (Entry for "memory.")

The scope and content of the prior art

11. Archibald discloses a method of charging for distributed software that is "based on the usage of the software as opposed to distribution sale." Col. 7, ll. 30-31.
12. Archibald uses a meter module as a means for determining the usage of the software. This is illustrated in Fig. 6 ("22 or 26").
13. "The meter module detects when a digital application is activated when the user of a user system requests the particular digital application from a server or initiates a run command for the digital application." Archibald, col. 11, ll. 38-42. *See also* Fig. 7.
14. Knapton, III is directed to object oriented programming and describes a method of controlling the usage of a software component by an application program on an end user's computer through the use of a password created from two keys, one created from an identifier of the application program and the other created from an identifier of the software component to be controlled. Col. 2, ll. 24-29.
15. Sobeski discloses the licensing of Java objects, which are software components. Col. 1, ll. 16-17.
16. Sobeski describes a method whereby a Java object is utilized only in accordance with terms specified in a license file. A license manager running on an end user's computer manages the use of the Java

- objects and ensures the use of the Java objects corresponds with the terms in the license. Col. 1, ll. 47-59.
17. Col. 6, l. 47 - col. 7, l. 60 of Sobeski describes an embodiment (shown in Fig. 3) whereby the Java object is “called” to verify that the object is licensed for use on the client computer. Sobeski here presents an example code for calling the license manager to determine if the Java object may be used on the client computer. It is further explained that once the license manager has been called, a license file associated with the Java object is received. Sobeski here describes an example of a format for the license file.
18. Sobeski describes Java objects, which can be a class of objects (col. 5, ll. 20), and states that “any number of instantiations of the object may be run on the client machine” (col. 5, ll. 21-22).
19. It is common knowledge that computers necessarily comprise a primary storage (i.e., “memory”) and that the primary storage is the component where programs reside.

PRINCIPLES OF LAW

Obviousness

“Section 103 forbids issuance of a patent when ‘the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.’” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art,

(2) any differences between the claimed subject matter and the prior art, and
(3) the level of skill in the art. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). *See also KSR*, 550 U.S. at 407 (“While the sequence of these questions might be reordered in any particular case, the [*Graham*] factors continue to define the inquiry that controls.”) The Court in *Graham* further noted that evidence of secondary considerations “might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented.” 383 U.S. at 17-18.

ANALYSIS

The rejection of claims 1-3, 5-7, 9-12, 14-25, 27-35, 37-40, 42-44, 53-55, and 74-77 under § 103(a) as being unpatentable over Archibald, Knapton, III, and Sobeski.

We will reverse this rejection as to claims 1-3, 5-7, 9-12, 14-25, 27-35, 37-40, 42-44, and 74-77.

All the claims 1-3, 5-7, 9-12, 14-25, 27-35, 37-40, 42-44, and 74-77 require the step of charging customers according to the use of distributed software modules wherein the software usage “is measured according to object instances created from the at least one object class.” Accordingly, to meet the initial burden of establishing a prima facie case of obviousness, the Examiner must show that this step is disclosed in the references cited against the claims or explain why one of ordinary skill in the art would have arrived at the claimed method comprising this step given what is disclosed or suggested by the cited references. *In re Oetiker*, 977 F.2d 1443, 1445 (Fed. Cir. 1992).

The Examiner repeated that “Sobiski [*sic.*, Sobeski] discloses ...

software usage [that] is measured according to object instances created from the at least one object class (column 6, line 47-column 7, line 60.).” Answer 4 and 15. We have reviewed the passage in Sobeski at column 6, line 47 - column 7, line 60. We have been unable to find there any discussion or suggestion of measuring software usage according to object instances created from an object class as the Examiner has argued. FF 17. Given the broadest reasonable construction of the claims in light of the Specification as it would be interpreted by one of ordinary skill in the art at the time of the invention, the step of charging customers according to the use of distributed software modules wherein the software usage “is measured according to object instances created from the at least one object class” broadly reads on charging customers for distributed programs (FF 4) according to their use, based on measuring the copies (i.e., “instances,” FF 7) made from the programs (“object classes,” FF 6). Even in light of this broad construction, we are unable to find the step at issue in Sobeski as the Examiner has argued. The passage in Sobeski relied upon describes instead a technique for controlling the use of Java objects on a client computer that involves determining if the Java object’s use matches terms specified in a license file. We see nothing there about creating instances (e.g., copies (*see* FF 7)) of an object class (e.g., a program (*see* FF 4)) as the claimed method requires, let alone measuring the usage of software based on object instances created from an object class.

As we explain in more detail below, Sobeski does in fact describe instantiating Java objects, which are a class of objects; thus disclosing the creating of instances. *See* FF 18. But the question here is not whether the prior art discloses simply the creating of instances but whether the prior

discloses or would have led one of ordinary skill in the art to a method having the step of “charging the customer according to use of the distributed software modules as determined during the monitoring step, wherein software usage is measured according to object instances created from the at least one object class” (claim 1). Even if the Examiner had pointed instead to the Sobeski disclosure of instantiating Java objects as evidence that the claim step at issue was disclosed, we would still disagree that Sobeski discloses the charging step at issue.

Also, irrespective of what the Examiner relied upon or could have relied upon in Sobeski, the Answer provides no reasoning as to why or how one of ordinary skill would have arrived at the claimed method given the disclosures of the cited references as a whole. “[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *KSR.*, 550 U.S. at 418 (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)).

Accordingly, we find that the Appellant has shown that a prima face case of obviousness of the subject matter of the method claims over Archibald, Knapton, III, and Sobeski has not been established.

We will affirm this rejection as to claims 53-55, for a number of reasons.

The focus of the Briefs was to challenge the Examiner’s position that the cited prior art does not render obvious the step of charging customers according to the use of distributed software modules wherein the software usage “is measured according to object instances created from the at least one object class.” That challenge does not apply to claims 53-55, which are

directed to a “memory” instead. In that regard, the Appellant simply argues that “Appellant traverses the rejection of claims 53-55 as obvious for at least the reason that Archibald does not disclose a single module containing the recited object class.” App. Br. 14. We do not find this argument persuasive as to error in the rejection. While the Examiner did find that Archibald discloses the subject matter of claims 53-55 (Answer 12-13), the statement of the rejection makes clear that these claims were rejected over the combination of Archibald, Knapton, III, and Sobeski. Accordingly, the Appellant’s challenge should have included addressing Knapton, III, and Sobeski.

As a matter of fact, we find that Sobeski, alone, discloses the claimed “memory.”

Claims 53-55 describe a memory containing a software module structure comprising (a) a supplier identification, (b) a product description, (c) a billing definition, and (d) an executable program module.

Limitations (a)-(c) are of no patentable consequence. These limitations represent information and, as such, are nonfunctional descriptive material. Nonfunctional descriptive material cannot render nonobvious an invention that would have otherwise been obvious. *In re Ngai*, 367 F.3d 1336, 1339 (Fed. Cir. 2004). *Cf. In re Gulack*, 703 F.2d 1381, 1385 (Fed. Cir. 1983) (when descriptive material is not functionally related to the substrate, the descriptive material will not distinguish the invention from the prior art in terms of patentability). Patentable weight need not be given to descriptive material absent a new and unobvious functional relationship between the descriptive material and the substrate. *See In re Lowry*, 32 F.3d 1579, 1582-83 (Fed. Cir. 1994); *In re Ngai*, 367 F.3d at 1338. *See also Ex*

parte Mathias, 84 USPQ2d 1276 (BPAI 2005) (nonprecedential) (Federal Circuit Appeal No. 2006-1103; WL 2433879, affirmed without written opinion Aug. 17, 2006). In that regard, we do not see a new and unobvious functional relationship between (a) a supplier identification, (b) a product description, and (c) a billing definition and a memory that would lead us to conclude that limitations (a)-(c) are consequential to the patentability of the claimed “memory.” We reach the same conclusion for the particular billing definitions that are recited in dependent claims 54 and 55.

Thus the question of patentability is reduced to whether the cited prior art renders obvious a “memory” comprising “an executable program module including at least one object class from which instantiated objects are rendered” (claim 53).

Given the ordinary and customary meaning of the terms “memory” (FF 10) and the definition provided in the Specification for the terms “executable program modules” (FF 5), “object class” (FF 6), and “objects” (FF 7), the broadest reasonable construction of the claim in light of the Specification as it would be interpreted by one of ordinary skill in the art is a primary storage of a computer containing a program that can be “instantiated” (*see* FF 9 for an ordinary and customary meaning for this term).

Sobeski describes Java objects, which can be a class of objects (col. 5, ll. 20), and states that “any number of instantiations of the object may be run on the client machine” (col. 5, ll. 21-22). FF 18. Accordingly, Sobeski teaches a program (Java objects) that can be “instantiated.” It is common knowledge that computers necessarily comprise a primary storage (i.e., “memory”) and that the primary storage is the component where programs

reside. FF 19. Accordingly, in disclosing the running of a program (Java objects) on a client computer, to one of ordinary skill in the art Sobeski necessarily teaches a “memory” and that “memory” would contain the program (Java objects). Since Sobeski further describes the instantiation of the program (Java objects) that would reside in the “memory,” it follows that Sobeski describes a “memory” comprising “an executable program module including at least one object class from which instantiated objects are rendered” (claim 53).

For the foregoing reasons, we affirm the rejection of claims 53-55.

We are cognizant that we have relied principally on Sobeski to affirm the claims and thus relied on less than all the references the Examiner applied in rejecting these claims. However, the Appellant has had an opportunity to review Knapton, III, in challenging the rejection of these claims but chose not to address it. We will not therefore denominate this as a new grounds of rejection. *Cf. Ex parte Sami Chemicals and Extracts, Ltd.* (BPAI 2009) (The informative status of this decision is noted at http://www.uspto.gov/web/offices/dcom/bpai/informative_opinions.html.) (The basic thrust of the rejection by the Examiner and the Board was the same, and we believe that Appellant has had a fair opportunity to react to those rejections.)

The rejection of claims 4, 8, 26, 36, and 41 under § 103(a) as being unpatentable over Archibald, Knapton, III, Sobeski, and Ahmad.

Claims 4, 8, 26, 36, and 41 depend from claims 3, 7, 1, 35, 37, respectively, whose rejection we have reversed above. For the same reasons we will reverse the Examiner’s rejection of these claims. *Cf. In re Fritch*,

972 F.2d 1260, 1266 (Fed. Cir. 1992) ("[D]ependent claims are nonobvious if the independent claims from which they depend are nonobvious.").

CONCLUSIONS

We conclude that the Appellant has shown that the Examiner erred in rejecting claims 1-3, 5-7, 9-12, 14-25, 27-35, 37-40, 42-44, and 74-77 under § 103(a) as being unpatentable over Archibald, Knapton, III, and Sobeski; and claims 4, 8, 26, 36, and 41 under § 103(a) as being unpatentable over Archibald, Knapton, III, Sobeski, and Ahmad. We conclude that the Appellant has not shown that the Examiner erred in rejecting claims 53-55 under § 103(a) as being unpatentable over Archibald, Knapton, III, and Sobeski.

DECISION

The decision of the Examiner to reject claims 1-12, 14-44, and 74-77 is REVERSED.

The decision of the Examiner to reject claims 53-55 is AFFIRMED.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv) (2007).

AFFIRMED-IN-PART

Appeal 2009-000451
Application 09/418,943

hh

MARK JOY
LEYDIG VOIT & MAYER LTD
TWO PRUDENTIAL PLAZA SUITE 4900
180 NORTH STETSON
CHICAGO, IL 60601-6780